

Microprocessor and Computer Architecture

UE21CS251B

4th Semester, Academic Year 2022-23

Date:

Name: Nikhil Girish	SRN: PES2UG21CS334	Section: F
---------------------	-----------------------	------------

Week# 2 Program Number: 1

Title of the Program

Write a program in ARM7TDMI-ISA to copy a block of N data items from Location A to Location B.

- a. Use Full word (.word directive)**
- b. Use Half word(.hword directive)**
- c. Use Byte wise (.Byte directive)**

I. ARM Assembly Code:

a:

```
.data
a: .word 10,20,30,40,50,60,70,80,90,100
b: .word 0,0,0,0,0,0,0,0,0,0

.text
LDR R0,=a
LDR R1,=b
MOV R4,#0
li:
```

```

LDR R3,[R0]
STR R3,[R1]
ADD R0,R0,#4
ADD R1,R1,#4
ADD R4,R4,#1
CMP R4, #10
BNE l1
SWI 0x011

.end

```

b:

```

.data
a: .hword 1,2,3,4,5,6,7,8,9,10
b: .hword 0,0,0,0,0,0,0,0,0,0

.text
LDRH R0,=a
LDRH R1,=b
MOV R4,#1
l1:
LDRH R3,[R0]
STRH R3,[R1]
ADD R0,R0,#2
ADD R1,R1,#2
ADD R4,R4,#1
CMP R4, #11
BNE l1

SWI 0x011

```

c:

```

.data
a: .byte 1,2,3,4,5,6,7,8,9,10
b: .byte 0,0,0,0,0,0,0,0,0,0

.text
LDR R0,=a
LDR R1,=b
MOV R4,#1

```

```

l1:
LDRB R3,[R0]
STRB R3,[R1]
ADD R0,R0,#1
ADD R1,R1,#1
ADD R4,R4,#1
CMP R4, #11
BNE l1

SWI 0x011

```

II. Output Screen Shots (Three)

The output should be verified for word, half word, byte

Word:

The screenshot displays a debugger interface with two main panes: 'RegistersView' on the left and 'CodeView' on the right.

RegistersView: The 'General Purpose' tab is active. It shows 16 registers (R0-R15) with their current values in hexadecimal. R0 is 00001050, R1 is 00001084, R2 is 00000000, R3 is 00000064, R4 is 0000000a, R5 is 00000000, R6 is 00000000, R7 is 00000000, R8 is 00000000, R9 is 00000000, R10 (sl) is 00000000, R11 (fp) is 00000000, R12 (ip) is 00000000, R13 (sp) is 00011400, R14 (lr) is 00000000, and R15 (pc) is 00001028. Below the registers, the CPSR Register status is shown: Negative (N) : 0, Zero (Z) : 1, Carry (C) : 1, Overflow (V) : 0, IRQ Disable : 1, FIQ Disable : 1, Thumb (T) : 0, and CPU Mode : System.

CodeView: The 'P1.o' file is open. It shows assembly code with addresses and disassembly. The code includes data sections for 'a' (word 10,20,30,40,50,60,70,80,90,100) and 'b' (word 0,0,0,0,0,0,0,0,0,0), and a text section. The assembly code is as follows:

```

00001034:0000000A    a: .word 10,20,30,40,50,60,70,80,90,100
:00000014
:0000001E
:00000028
:00000032
0000105C:00000000    b: .word 0,0,0,0,0,0,0,0,0,0
:00000000
:00000000
:00000000
:00000000

:00001000:E59F0024    LDR R0,=a
00001004:E59F1024    LDR R1,=b
00001008:E3A04000    MOV R4,#0
l1:
0000100C:E5903000    LDR R3,[R0]
00001010:E5813000    STR R3,[R1]
00001014:E2800004    ADD R0,R0,#4
00001018:E2811004    ADD R1,R1,#4
0000101C:E2844001    ADD R4,R4,#1
00001020:E354000A    CMP R4, #10
00001024:1AFFFFF8    BNE l1
00001028:EF000011    SWI 0x011

0000102C:00000000    .end...
:00000028

```

The 'WatchView' pane at the bottom is currently empty.

Half-word:

RegistersView

General Purpose Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 00001048

R1 : 0000105c

R2 : 00000000

R3 : 0000000a

R4 : 0000000b

R5 : 00000000

R6 : 00000000

R7 : 00000000

R8 : 00000000

R9 : 00000000

R10 (s1) : 00000000

R11 (fp) : 00000000

R12 (ip) : 00000000

R13 (sp) : 00011400

R14 (lr) : 00000000

R15 (pc) : 00001028

CPSR Register

Negative (N) : 0

Zero (Z) : 1

Carry (C) : 1

Overflow (V) : 0

IRQ Disable : 1

FIQ Disable : 1

Thumb (T) : 0

CPU Mode : System

CodeView

P1.o

.data

00001034:00020001 a: .hword 1,2,3,4,5,6,7,8,9,10

:00040003

:00060005

:00080007

:000A0009

00001048:00000000 b: .hword 0,0,0,0,0,0,0,0,0,0

:00000000

:00000000

:00000000

:00000000

:00000000

.text

00001000:E1DF02B4 LDRH R0,=a

00001004:E1DF12B4 LDRH R1,=b

00001008:E3A04001 MOV R4,#1

l1:

0000100C:E1D030B0 LDRH R3,[R0]

00001010:E1C130B0 STRH R3,[R1]

00001014:E2800002 ADD R0,R0,#2

00001018:E2811002 ADD R1,R1,#2

0000101C:E2844001 ADD R4,R4,#1

00001020:E354000B CMP R4, #11

00001024:1AFFFFFF8 BNE l1

00001028:EF000011 SWI 0x011...

:00000000

:00000014

Byte:

RegistersView

General Purpose Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 0000103e

R1 : 00001048

R2 : 00000000

R3 : 0000000a

R4 : 0000000b

R5 : 00000000

R6 : 00000000

R7 : 00000000

R8 : 00000000

R9 : 00000000

R10 (s1) : 00000000

R11 (fp) : 00000000

R12 (ip) : 00000000

R13 (sp) : 00011400

R14 (lr) : 00000000

R15 (pc) : 00001028

CPSR Register

Negative (N) : 0

Zero (Z) : 1

Carry (C) : 1

Overflow (V) : 0

IRQ Disable : 1

FIQ Disable : 1

Thumb (T) : 0

CPU Mode : System

CodeView

P1.o

.data

00001034:04030201 a: .byte 1,2,3,4,5,6,7,8,9,10

:08070605

:0A09

0000103E:00000000 b: .byte 0,0,0,0,0,0,0,0,0,0

:00000000

:0000

.text

00001000:E59F0024 LDR R0,=a

00001004:E59F1024 LDR R1,=b

00001008:E3A04001 MOV R4,#1

l1:

0000100C:E5D03000 LDRB R3,[R0]

00001010:E5C13000 STRB R3,[R1]

00001014:E2800001 ADD R0,R0,#1

00001018:E2811001 ADD R1,R1,#1

0000101C:E2844001 ADD R4,R4,#1

00001020:E354000B CMP R4, #11

00001024:1AFFFFFF8 BNE l1

00001028:EF000011 SWI 0x011...

:00000000

:0000000A

Microprocessor and Computer Architecture

UE21CS251B

4th Semester, Academic Year 2022-23

Date:

Name: Nikhil Girish	SRN: PES2UG21CS334	Section: F
---------------------	-----------------------	---------------

Week# ____2____ Program Number: ____2____

Title of the Program

Write a program in ARM7TDMI-ISA to find the sum of N data items in the memory. Store the result in the memory location.

- a. Use Full word (.word directive)**
- b. Use Half word(.hword directive)**
- c. Use Byte wise (.Byte directive)**

I.ARM Assembly Code

a:

```
.data
a: .word 10,20,30,40,50,60,70,80,90,100
b: .word 0

.text
LDR R0,=a
LDR R1,=b
MOV R4,#0
```

```

MOV R5,#0

l:
LDR R2,[R0]
LDR R3,[R1]
ADD R4,R2,R3
STR R4,[R1]
ADD R0,R0,#4
ADD R5,R5,#1
CMP R5, #10
BNE l

SWI 0x011

```

b:

```

.data
a: .hword 10,20,30,40,50,60,70,80,90,100
b: .hword 0

.text
LDR R0,=a
LDR R1,=b
MOV R4,#0
MOV R5,#0

l:
LDRH R2,[R0]
LDRH R3,[R1]
ADD R4,R2,R3
STRH R4,[R1]
ADD R0,R0,#2
ADD R5,R5,#1
CMP R5, #10
BNE l

SWI 0x011

```

c:

```

.data
a: .byte 10,20,30,40,50,60,70,80,90,100
b: .byte 0

```

```

.text
LDR R0,=a
LDR R1,=b
MOV R4,#0
MOV R5,#0

l:
LDRB R2,[R0]
LDRB R3,[R1]
ADD R4,R2,R3
STRB R4,[R1]
ADD R0,R0,#1
ADD R5,R5,#1
CMP R5, #10
BNE l
SWI 0x011

```

II. Output Screen Shots (Three)

The output should be verified for word, half word, byte

Word:

The screenshot shows a debugger interface with two main panes. The left pane, titled 'RegistersView', displays the state of various registers. The right pane, titled 'CodeView', shows the assembly code being executed.

RegistersView:

- General Purpose: Floating Point
- Hexadecimal
- Unsigned Decimal
- Signed Decimal
- R0: 00001064
- R1: 00001064
- R2: 00000064
- R3: 000001c2
- R4: 00000226
- R5: 0000000a
- R6: 00000000
- R7: 00000000
- R8: 00000000
- R9: 00000000
- R10 (s1): 00000000
- R11 (fp): 00000000
- R12 (ip): 00000000
- R13 (sp): 00011400
- R14 (lr): 00000000
- R15 (pc): 00001030
-
- CPSR Register
- Negative (N): 0
- Zero (Z): 1
- Carry (C): 1
- Overflow (V): 0
- IRQ Disable: 1
- FIQ Disable: 1
- Thumb (T): 0
- CPU Mode: System

CodeView:

```

P2.o
.data
0000103C:0000000A    a: .word 10,20,30,40,50,60,70,80,90,100
:00000014
:0000001E
:00000028
:00000032
00001064:00000000    b: .word 0

.text
00001000:E59F002C    LDR R0,=a
00001004:E59F102C    LDR R1,=b
00001008:E3A04000    MOV R4,#0
0000100C:E3A05000    MOV R5,#0

l:
00001010:E5902000    LDR R2,[R0]
00001014:E5913000    LDR R3,[R1]
00001018:E0824003    ADD R4,R2,R3
0000101C:E5814000    STR R4,[R1]
00001020:E2800004    ADD R0,R0,#4
00001024:E2855001    ADD R5,R5,#1
00001028:E355000A    CMP R5, #10
0000102C:1AFFFFF7    BNE l
00001030:EF000011    SWI 0x011...
:00000000
:00000028

```

Half-word:

RegistersView

General Purpose Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 00001050

R1 : 00001050

R2 : 00000064

R3 : 000001c2

R4 : 00000226

R5 : 0000000a

R6 : 00000000

R7 : 00000000

R8 : 00000000

R9 : 00000000

R10 (s1) : 00000000

R11 (fp) : 00000000

R12 (ip) : 00000000

R13 (sp) : 00011400

R14 (lr) : 00000000

R15 (pc) : 00001030

CPSR Register

Negative (N) : 0

Zero (Z) : 1

Carry (C) : 1

Overflow (V) : 0

IRQ Disable : 1

FIQ Disable : 1

Thumb (T) : 0

CPU Mode : System

CodeView

P2.o

.data

0000103C:0014000A a: .hword 10,20,30,40,50,60,70,80,90,100

:0028001E

:003C0032

:00500046

:0064005A

00001050:0000 b: .hword 0

.text

00001000:E59F002C LDR R0,=a

00001004:E59F102C LDR R1,=b

00001008:E3A04000 MOV R4,#0

0000100C:E3A05000 MOV R5,#0

l:

00001010:E1D020B0 LDRH R2,[R0]

00001014:E1D130B0 LDRH R3,[R1]

00001018:E0824003 ADD R4,R2,R3

0000101C:E1C140B0 STRH R4,[R1]

00001020:E2800002 ADD R0,R0,#2

00001024:E2855001 ADD R5,R5,#1

00001028:E355000A CMP R5, #10

0000102C:1AFFFFF7 BNE l

00001030:EF000011 SWI 0x011...

:00000000

:00000014

Byte:

RegistersView

General Purpose Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 00001046

R1 : 00001046

R2 : 00000064

R3 : 000000c2

R4 : 00000126

R5 : 0000000a

R6 : 00000000

R7 : 00000000

R8 : 00000000

R9 : 00000000

R10 (s1) : 00000000

R11 (fp) : 00000000

R12 (ip) : 00000000

R13 (sp) : 00011400

R14 (lr) : 00000000

R15 (pc) : 00001030

CPSR Register

Negative (N) : 0

Zero (Z) : 1

Carry (C) : 1

Overflow (V) : 0

IRQ Disable : 1

FIQ Disable : 1

Thumb (T) : 0

CPU Mode : System

CodeView

P2.o

.data

0000103C:281E140A a: .byte 10,20,30,40,50,60,70,80,90,100

:50463C32

:645A

00001046:00 b: .byte 0

.text

00001000:E59F002C LDR R0,=a

00001004:E59F102C LDR R1,=b

00001008:E3A04000 MOV R4,#0

0000100C:E3A05000 MOV R5,#0

l:

00001010:E5D02000 LDRB R2,[R0]

00001014:E5D13000 LDRB R3,[R1]

00001018:E0824003 ADD R4,R2,R3

0000101C:E5C14000 STRB R4,[R1]

00001020:E2800001 ADD R0,R0,#1

00001024:E2855001 ADD R5,R5,#1

00001028:E355000A CMP R5, #10

0000102C:1AFFFFF7 BNE l

00001030:EF000011 SWI 0x011...

:00000000

:0000000A

Microprocessor and Computer Architecture

UE21CS251B

4th Semester, Academic Year 2022-23

Date:

Name: Nikhil Girish	SRN: PES2UG21CS334	Section: F
---------------------	-----------------------	------------

Week# ____2____ Program Number: ____3____

Title of the Program

Write a program in ARM7TDMI-ISA to find the sum of N natural numbers. Store the result in the memory location.

I.ARM Assembly Code:

```
.data
a: .byte 1,2,3,4,5,6,7,8,9,10
b: .byte 0

.text
LDR R0,=a
LDR R1,=b
MOV R4,#0
MOV R5,#0

l:
LDRB R2,[R0]
LDRB R3,[R1]
ADD R4,R2,R3
STRB R4,[R1]
ADD R0,R0,#1
ADD R5,R5,#1
CMP R5, #10
```

```
BNE 1
SWI 0x011
```

II. Output Screen Shots (One):

The screenshot displays a debugger interface with two main panes: **RegistersView** on the left and **CodeView** on the right.

RegistersView: The **General Purpose** tab is selected. The **Hexadecimal** view is chosen. The registers R0 through R15 are listed with their values. R15 (PC) is highlighted in red. Below the registers, the **CPSR Register** is shown with the following flags: Negative (N): 0, Zero (Z): 1, Carry (C): 1, Overflow (V): 0, IRQ Disable: 1, FIQ Disable: 1, Thumb (T): 0, and CPU Mode: System.

CodeView: The **P3.o** file is open. The **.data** section contains two byte arrays: `a: .byte 1,2,3,4,5,6,7,8,9,10` and `b: .byte 0`. The **.text** section contains assembly instructions: `LDR R0,=a`, `LDR R1,=b`, `MOV R4,#0`, `MOV R5,#0`, `LDRB R2,[R0]`, `LDRB R3,[R1]`, `ADD R4,R2,R3`, `STRB R4,[R1]`, `ADD R0,R0,#1`, `ADD R5,R5,#1`, `CMP R5,#10`, and `BNE 1`. The instruction `SWI 0x011...` is highlighted in blue.

Microprocessor and Computer Architecture

UE21CS251B

4th Semester, Academic Year 2022-23

Date:

Name: Nikhil Girish	SRN: PES2UG21CS334	Section: F
---------------------	-----------------------	------------

Week# ____2____

Program Number: ____4____

Title of the Program

Write a program in ARM7TDMI-ISA to find the product of two 32bit numbers using barrel shifter.

I.ARM Assembly Code:

```
.data
a: .word -4
b: .word 9

.text
LDR R0, =a
LDR R0, [R0]
LDR R1, =b
LDR R1, [R1]
MOV R2, #0
MOV R3, #0

l:
    CMP R0, #0
    BEQ end
    TST R0, #1
    BEQ l1
    ADD R3, R3, R1, LSL R2

l1:
```

```

MOV R0, R0, LSR #1
ADD R2, R2, #1
B 1

end:
SWI 0x011

```

II. Output Screen Shot (One):

The screenshot displays a debugger interface with two main panes: **RegistersView** and **CodeView**.

RegistersView: The **General Purpose** tab is active, showing the state of 16 registers (R0-R15) in hexadecimal. R15 (pc) is highlighted in red and shows the value 00001038. Below the registers, the **CPSR Register** status is shown:

- Negative (N) : 0
- Zero (Z) : 1
- Carry (C) : 1
- Overflow (V) : 0
- IRQ Disable: 1
- FIQ Disable: 1
- Thumb (T) : 0
- CPU Mode : System

CodeView: The **P4.o** file is loaded, showing assembly code. The **.data** section contains two words: 'a' at address 00001044 (value FFFFFFFC) and 'b' at address 00001048 (value 00000009). The **.text** section contains the main code:

```

00001000:E59F0034 LDR R0, =a
00001004:E5900000 LDR R0, [R0]
00001008:E59F1030 LDR R1, =b
0000100C:E5911000 LDR R1, [R1]
00001010:E3A02000 MOV R2, #0
00001014:E3A03000 MOV R3, #0

l1:
00001018:E3500000 CMP R0, #0
0000101C:0A000005 BEQ end
00001020:E3100001 TST R0, #1
00001024:0A000000 BEQ l1
00001028:E0833211 ADD R3, R3, R1, LSL R2

l1:
0000102C:E1A000A0 MOV R0, R0, LSR #1
00001030:E2822001 ADD R2, R2, #1
00001034:EAF00007 B 1

end:
00001038:EF000011 SWI 0x011...

```

The instruction at address 00001038 (SWI 0x011...) is highlighted in blue.

Microprocessor and Computer Architecture

UE21CS251B

4th Semester, Academic Year 2022-23

Date:

Name: Nikhil Girish	SRN: PES2UG21CS334	Section: F
---------------------	-----------------------	------------

Week# ____2____

Program Number: ____5____

Title of the Program

Convert the following statement in C language into an ALP using ARM7TDMI – ISA.

IF([A]==[B]) then C=[A]+[B];

ELSE IF ([B]==[C]) D=[A]-[B];

ELSE E=[A]*[B]

Where A,B C, D & E are memory locations.

I.ARM Assembly Code:

```
.data
```

```
a: .word 2
b: .word 3
c: .word 4
d: .word 0
e: .word 0

.text
LDR R0, =a
LDR R1, =b
LDR R2, =c
LDR R3, =d
LDR R4, =e

LDR R0, [R0]
LDR R1, [R1]
LDR R5, [R2]

TEQ R0, R1
BEQ l
TEQ R1, R5
BEQ l1

MUL R6, R0, R1
STR R6, [R4]
B end

l:
    ADD R6, R0, R1
    STR R6, [R2]
    B end
l1:
    SUB R6, R0, R1
    STR R6, [R3]
    B end

end:
SWI 0x011
```

II. Output Screen Shot (One):

RegistersView

General Purpose

Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 00000002

R1 : 00000003

R2 : 00001074

R3 : 00001078

R4 : 0000107c

R5 : 00000004

R6 : 00000006

R7 : 00000000

R8 : 00000000

R9 : 00000000

R10 (sl) : 00000000

R11 (fp) : 00000000

R12 (ip) : 00000000

R13 (sp) : 00011400

R14 (lr) : 00000000

R15 (pc) : 00001054

CPSR Register

Negative (N) : 0

Zero (Z) : 0

Carry (C) : 0

Overflow (V) : 0

IRQ Disable : 1

FIQ Disable : 1

Thumb (T) : 0

CPU Mode : System

CodeView

P5.o

```

.data
0000106C:00000002    a: .word 2
00001070:00000003    b: .word 3
00001074:00000004    c: .word 4
00001078:00000000    d: .word 0
0000107C:00000000    e: .word 0

.text
00001000:E59F0050    LDR R0, =a
00001004:E59F1050    LDR R1, =b
00001008:E59F2050    LDR R2, =c
0000100C:E59F3050    LDR R3, =d
00001010:E59F4050    LDR R4, =e

00001014:E5900000    LDR R0, [R0]
00001018:E5911000    LDR R1, [R1]
0000101C:E5925000    LDR R5, [R2]

00001020:E1300001    TEQ R0, R1
00001024:0A000004    BEQ 1
00001028:E1310005    TEQ R1, R5
0000102C:0A000005    BEQ 11

00001030:E0060190    MUL R6, R0, R1
00001034:E5846000    STR R6, [R4]
00001038:EA000005    B end

1:
0000103C:E0806001    ADD R6, R0, R1
00001040:E5826000    STR R6, [R2]
00001044:EA000002    B end

```

Microprocessor and Computer Architecture

UE21CS251B

4th Semester, Academic Year 2022-23

Date:

Name: Nikhil Girish	SRN: PES2UG21CS334	Section: F
---------------------	-----------------------	------------

Week# ____2____

Program Number: ____6__

Title of the Program

Write a program in ARM7TDMI-ISA to find the factorial of a number.

I.ARM Assembly Code:

```
.data
a: .word 7

.text
LDR R0,=a
LDR R0, [R0]
MOV R1,#1
l:
    MUL R1,R0,R1
    SUBS R0,R0,#1
    BNE l
SWI 0x011
```


II. Output Screen Shot (One):

The screenshot displays a debugger interface with two main panes: **RegistersView** on the left and **CodeView** on the right.

RegistersView:

- General Purpose registers are shown in hexadecimal format.
- Registers R0 through R15 are listed with their values. R15 (PC) is highlighted in red.
- CPSR Register status is shown below the general purpose registers.

Register	Value
R0	00000000
R1	000013b0
R2	00000000
R3	00000000
R4	00000000
R5	00000000
R6	00000000
R7	00000000
R8	00000000
R9	00000000
R10 (s1)	00000000
R11 (fp)	00000000
R12 (ip)	00000000
R13 (sp)	00011400
R14 (lr)	00000000
R15 (pc)	00001018

CPSR Register

Negative (N)	:0
Zero (Z)	:1
Carry (C)	:1
Overflow (V)	:0
IRQ Disable	:1
FIQ Disable	:1
Thumb (T)	:0
CPU Mode	:System

CodeView:

File: P6.o

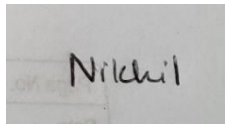
```
.data
00001020:00000007      a: .word 7

.text
00001000:E59F0014      LDR R0,=a
00001004:E5900000      LDR R0, [R0]
00001008:E3A01001      MOV R1,#1
l:
0000100C:E0010190      MUL R1,R0,R1
00001010:E2500001      SUBS R0,R0,#1
00001014:1AFFFFFC      BNE l
00001018:EF000011      SWI 0x011...
:00000000
```

Disclaimer:

- The programs and output submitted is duly written, verified and executed by me.
- I have not copied from any of my peers nor from the external resource such as internet.
- If found plagiarized, I will abide with the disciplinary action of the University.

Signature:

A rectangular box containing a handwritten signature in black ink. The signature appears to be 'Nikhil'.

Name: Nikhil Girish

SRN: PES2UG21CS334

Section: 4F

Date: 28-01-2023